Lecture 5

dinsdag 29 november 2022 10:37

secure communication

- confidentiality
- authenticity
  - → (integrity)

application	needs changes to every application... as well as key management,...	PGP, S/MIME, OTR	→ only gives point-to-point security (not end-to-end, in that sense)
transport	encrypt sessions / messages (client-server)	TLS/SSL	→ no encryption between front-end and application servers
network	encrypt IP packets	IPSec, Wireguard	
link	data security between endpoints → point-to-point security	WPA2	
physical layer			

IPSec packet:

header
payload

provided functionality:

- packet integrity
- packet origin authentication
- confidentiality
- traffic flow confidentiality
- replay attack protection

transport mode: only payload is protected, header is not encrypted in ESP, parts of header and entire payload are authenticated
only used between hosts

tunnel mode: header and data are protected; put (encrypted) into new packet; provides data flow confidentiality (to some extent)
typically used for VPN

authentication header (AH):

- next header (type)
- payload length (of AH)
- security parameter index (SPI)
- sequence number — against replay attacks
- authentication data (MAC)

ESP packet contains

- SPI
- sequence number
- payload data
- padding
- padding length
- next header
- authentication data

security associations: parameters

- sequence number, overflow
- anti-replay window
- ⋮

security policy database: describes what security associations are supported by certain points

should do both encryption & authentication

many key-agreement protocols exist

---

key exchange security requirements

- confidentiality
- authenticity — of at least one party
- correctness
- forward secrecy — confidentiality remains if one party's keys are compromised
- post-compromise security — re-establish security after compromise

AE = authenticated encryption

AD = associated data

KEM = key encapsulation method

AKE = authenticated key exchange

FS-AKE = forward-secure AKE

↳ add ephemeral key so that a different, unknown key is used for every session; sign ephemeral key with 'usual' key
↓
since this is destroyed after session, recovery of plaintext becomes impossible after session

es → initiator has ephemeral key
responder has static key

h = handshake hash : hash of all handshake messages using hash

ck = chaining key : function of all DH results so far using HKDF

new ck ⇒ derive session key k & nonce n
↓
encrypt payload with k

N = no static key
K = known
X = transmitted
I = immediately transmitted

goals

- authenticity
  - none, no KCI, KCI
  - KCI = key compromise impersonation
- confidentiality
  - none, different levels of forward secrecy
- identity hiding

Wireguard: simple VPN protocol

noise with IK pattern

TLS = variant of SSL v3

- → designed for web, also used for mail/SMTP,...
- → provides transparency for higher (i.e. application) layer
- → runs on top of TCP

handshake: initiate session →

application: data transfer

alert: of exceptional conditions

client → server:

- clienthello →
- ← serverhello
- ← certificate
- ← serverkey-exchange
- (← certificate-request)
- ← server-hellodone
- clientkeyexchange →
- (certificate verify →)
- changecipherspec →
- finished →
- ← changecipherspec
- ← finished